Microsoft

Building the User Interface with HTML5

2.1. Choose and configure HTML5 tags to display text content.
2.2. Choose and configure HTML5 tags to display graphics.
2.3. Choose and configure HTML5 tags to play media.

Agenda

1. HTML
2. Basic Markup and Page Structure
3. Text Elements
4. Displaying Graphics
5. Canvas & SVG
6. Media in HTML5

HTML

Hypertext Markup Language (HTML)

```
<!DOCTYPE html>

<html lang="en">
    <head>
        <meta charset="utf-8" />
        <title>Basic HTML</title>
    </head>
    <body>
      <h1>This is a Basic
Header</h1>
      <p>This is a basic
paragraph.</p>
    </body>
</html>
```

- HTML is the language used to provide structure to web pages
- HTML uses tags, such as <p> and <h1> to perform this function
- Browsers “read” HTML files and then produce web pages based on the tags that are used

Versions of HTML

- Throughout the 2000s, HTML 4.01 was the standard for web pages
  - HTML 4.01 was limited in what it could provide users
- A strong demand for a rich web experience, including audio, video, and interactivity led to the development of a new HTML standard

HTML5

- The **World Wide Web Consortium (W3C)** is the standards organization responsible for the development of HTML5
- The HTML5 standard encompasses HTML markup tags, **Cascading Style Sheets (CSS)**, and **JavaScript**
- HTML5 is **platform-independent**

New with HTML5

FEATURE	DESCRIPTION
Audio and video tags	Embeds video on web pages using the <audio> and <video> tags
Canvas	Creates space for JavaScript to draw graphics on a web page
Media queries	A feature in CSS3 that detects screen size and adjusts output to fit
New application programming interfaces	Provides access to many digital resources that can be incorporated into the code of web applications
Geolocation	Uses JavaScript to detect the geographic location of a device

Basic Markup and Page Structure

HTML Tags

opening tag

```
<!DOCTYPE html>

<html lang="en">
    <head>
        <meta charset="utf-8" />
        <title>Basic HTML</title>
    </head>
    <body>
        <h1>This is a basic
header</h1>
        <p>This is a basic
paragraph.</p>
    </body>
</html>
```

closing tag

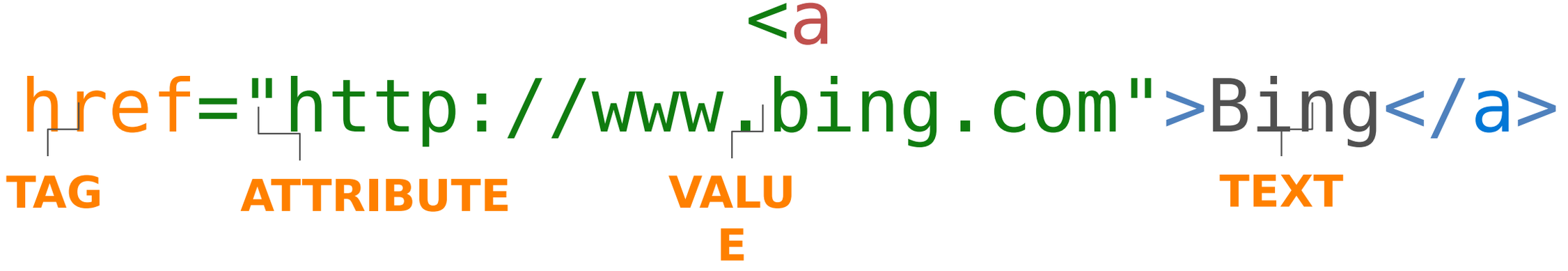
- A tag is a **keyword** surrounded by angle brackets
- Most tags come in pairs, with an opening tag and a closing tag
  - Closing tags are identical, but include a slash before the keyword
- A tag pair or an empty tag is called an **element**

Common HTML Tags

TAG	PURPOSE
`<html>`	identifies a page as an HTML document
`<head>`	contains code used by the browser to add interactivity or style a page
`<title>`	title of a document
`<body>`	surrounds content that is visible on a Web page
`<p>`	paragraphs
`<a href=“URL”>`	links
`<h1>`	top heading
`<img>`	images

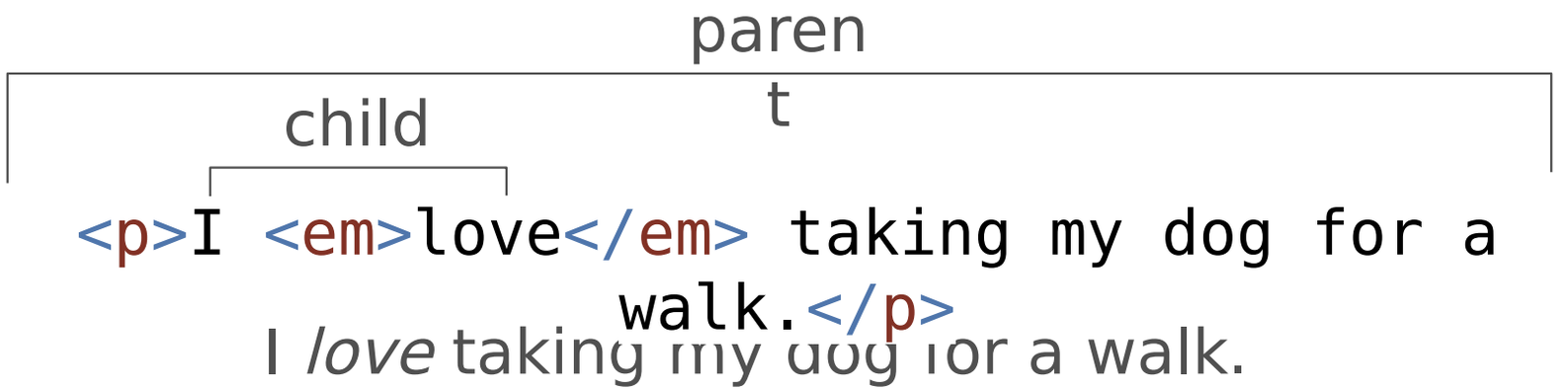
Using Attributes

- Tags are used in combination with **attributes** to describe how data should be rendered on a Web page
  - In other words, attributes can be used to provide additional information that tags cannot provide alone
- Each element has a specific set of attributes that can be used with it
  - HTML5 includes **global attributes**, which can be used with any element
- Attributes are added to tags using the following syntax:



Nesting Elements

- Creating awesome web pages requires you to combine elements, their attributes, and engaging content
- When two or more elements apply to the same block of content, then you have to nest tag pairs
- **Nesting** is the process of placing one element inside of another
  - The outside element is called a **parent**, while the inner element is called a **child**



Special Characters in HTML

- A special character, such as a percent sign or a copyright symbol, is known as an **entity** in HTML
- Including entities in a Web page requires character encoding or the special characters will not render
  - Users will see garbled text instead
- Each special character can be reproduced using its entity name OR a numerical code
  - Each entity starts with an ampersand (&) and ends with a semicolon (;)

SPECIAL CHARACTER	DESCRIPTION	ENTITY NAME	CODE
©	Copyright	&copy;	&#169;
$	Dollar Sign	&dollar;	&#36;
%	Percent Sign	&percnt;	&#37;
&	Ampersand	&amp;	&#38;

The DOCTYPE

- A **doctype declaration** is used to help a Web browser determine which rules it should use for rendering a Web page
- In HTML 4, doctype declarations require a reference to a Document Type Definition (DTD) and looks quite complex
- In HTML5, the doctype declaration is simpler, as shown below

HTML 4

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.1//EN"
"http://www.example.com/TR/xhtml11/DTD/
xhtml11.dtd">
```

HTML5

```
<!DOCTYPE html>
```

HTML5 Elements in Action

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <title>HTML5 Elements in Action</title>
    </head>
    <body>
        <h1>This is a header</h1>
        <p>This is a paragraph.</p>
    </body>
</html>
```

Text Elements

HTML Text Elements

ELEMENT	FUNCTIONALITY
<b>	Defines bold text
	Defines emphasized text
<i>	Defines italicized text
<small>	Defines smaller text
	Defines important text
<sub>	Defines subscript text
<sup>	Defines superscript text

Text Element Demo

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <title>HTML Example</title>
 </head>
 <body>
    <h1>This is a sample page.</h1>
    <p>This page includes <em>some</em> <strong>nested
    elements</strong>.</p>
 </body>
</html>
```

Text Elements with New Functionality

ELEMENT	FUNCTIONALITY IN HTML 4	FUNCTIONALITY IN HTML5
<b>	emphasize text by making it bold	“stylistically offsets” text
<i>	emphasize text by making it italic	“alternate voice or mood”
	N/A	labels text as strong importance, while making it appear bold
	N/A	indicates emphatic stress, while making it appear italic

Text Elements No Longer Used

- With the addition of new elements, W3C earmarks elements or attributes for removal
  - Elements and attributes are removed because they are no longer useful
- The process of removing elements is called **deprecation**
- Deprecated elements may still render in older browsers, but best practice suggests you should not use them if developing for newer

DEPRECATED ELEMENT	NEW ELEMENT
<acronym>	<abbr>
<applet>	<object>
<basefont>	Use CSS instead
<big>	Use CSS instead
	Use CSS instead
<dir>	
<font>	Use CSS instead
<strike>	<del> or CSS instead

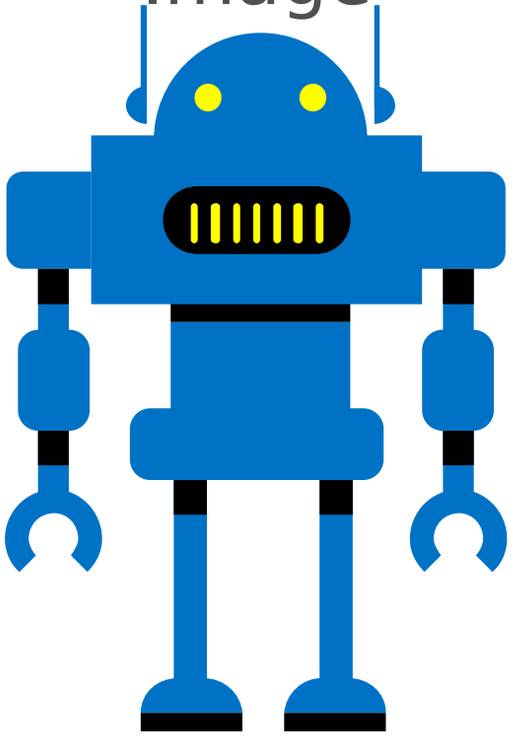
Displaying Graphics

Images and Graphics in HTML

- Images are an incredibly important aspect of creating engaging Web pages
- There are two major categories of images that can be used:
  - raster (bitmap)
  - vector
- **Raster images** are made up of pixels, while **vector images** are made of lines and curves

Raster Image

Vector Image



Raster vs. Vector Images

Raster Images

- Photographs are raster images
- Raster file formats include JPG, PNG, GIF, and BMP
- Raster images pixelate when they are enlarged

Vector Images

- Digital illustrations are typically vector images
- Vector images maintain quality when enlarged
- Vector images are typically made with advanced programs like Photoshop or CorelDRAW, then converted into PNG or GIF formats (raster file types)

`img` Element

- Add images to a Web page using the `<img>` tag
  - NOTE: A closing tag is not required for the image element
- The `<img>` tag requires use of the `src` and `alt` attributes
  - `src` stands for source
  - `alt` stands for alternative
- `src` defines the pathway for the image file, while the value of the `alt` attribute makes text accessible to people with disabilities

Attributes of the `img` element

ATTRIBUTE	VALUE	DESCRIPTION
`src`	URL	Specifies the location of an image
`alt`	Text	Specifies alternate text for an image, which displays when a user hovers their mouse pointer over it
`height`	pixels	Specifies the height of an image
`width`	pixels	Specifies the width of an image
`ismap`	ismap	Specifies an image as a server-side image map
`usemap`	#mapname	Specifies an image as a client-side image map

figure and figcaption Elements

- The img element can be used in combination with two new elements, figure and figcaption, to organize images and provide captions
- The figure element specifies the type of figure that is being added, and can also be used to group images side by side
- The figcaption element can be used to add captions before or after images

```
<figure>
    <figcaption>Who wouldn't want to take this dog for a walk?</figcaption>
    <img src="dog.jpg" alt="This is a picture of my dog, Stella."
title="Stella"/>
</figure>
```

Canvas & SVG

The canvas Element

- The canvas element creates a blank container for graphics
- It’s a new element in HTML5 and you can draw graphics using JavaScript
  - Drawing on a canvas is done by using the Canvas API
- Canvas can is used by developers to create 2D games or animations

```
<canvas id="myCanvas" width=“500px"
height=“300px"></canvas>
```

this code makes a blank container

The canvas Element in Action

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Canvas Test</title>
    <script>
        function f1() {
            var canvas = document.getElementById("smlRectangle");
            context = canvas.getContext("2d");
            context.fillStyle = "rgb(0,0,255)";
            context.fillRect(10, 20, 200, 100);
        }
    </script>
</head>
<body onload = "f1();">
    <canvas id="smlRectangle" height="100" width="200 "></canvas>
</body>
</html>
```

Alternative Images or Text for Older Browsers

- The canvas element won’t work on some older browsers
- To avoid problems with rendering, you should add content that will display when a canvas drawing cannot
- The content can be an image or text

```
<canvas id="smlRectangle2" height="100" width="200">
        <img

src="http://www.example.com/images/smlRectangle.jpg"
 alt="A blue rectangle" />
</canvas>
```

Scalable Vector Graphics

- Scalable Vector Graphics (SVG) is a language for describing 2D graphics in XML
- With SVG, you provide drawing instructions within your code versus using the src attribute to point to a file
- Unlike other image types, graphics created using the svg element are scalable
  - The quality of the image will not change if it shrinks or is enlarged

```
<svg height="1000px" width="1000px">
    <rect id="myRect" height="100px" width="100px"
fill="blue"/>
</svg>
```

Canvas vs. SVG

Canvas

- Use for small drawings
- Use for drawings with a lot of objects in them
- Use for small screens
- Use for real-time data output, such as maps or weather data

Scalable Vector Graphics

- Use for larger graphics
- Use for drawings with a small number of objects
- Use for drawings that require a large number of pixels
- Use for highly detailed vector graphics

Media in HTML5

Media in HTML5

- Multimedia is a key component of a high-quality Web browsing experience
- Before HTML5, browsers depended on plug-ins and media players to allow users to listen to music and watch videos
- Now, browsers that support HTML5 can provide access to multimedia with the `<video>` and `<audio>` tags

Video Tags

- You can embed video into HTML documents using the `<video>` element
- Use the source attribute to point to the location of the video file
- The `height` and `width` attributes help determine how the video will appear on a Web page

```
<video src=”cat_vid.mp4" height=“300” width=“400”></video>
```

Video Control Attributes

There are a number of other attributes that can be used with the video tag to add control of the video

poster	displays a static image before the video loads
autoplay	starts playing the video automatically upon page load
controls	displays controls for controlling the volume, playing, pausing, and stopping the video
loop	plays the video on repeat

HTML

```
<video
    src=“cat_vid.mp4
    width="400"
    height="300"
    poster=”meow.jpg"
    autoplay
    controls
    loop>
</video>
```

Video Formats

- A number of video formats are supported by Web browsers, including MP3, H.264, OGG, and WebM
- When you specify the video type, you should also specify the **codec**
  - A codec is the technology used for compressing data
- It's a best practice to use the `<source>` tag in combination with its `type` attribute to point to the file and specify the file type

HTML

```
<video
   width="400"
   height="300"
poster="meow.jpg"
autoplay="autoplay"
controls="controls"
loop="loop">
   <source
src="cat_vid.mp4"
type="video/mp4" />
</video>
```

Browser Compatibility

- Not all video formats are compatible with all browsers
- The MP4 format is the most commonly used by Web browsers and mobile devices
- To ensure that a video is compatible with a majority of browsers and devices, you should use the <source> element to include multiple formats

HTML

```
<video
   width="400"
   height="300"
   poster=“meow.jpg"
autoplay="autoplay"
controls="controls"
   loop="loop">

   <source src=”cat_vid.mp4"
type="video/mp4" />

   <source src=“cat_vid.ogg”
type=‘video/ogg;
   codecs=“theora, vorbis”’>
</video>
```

Audio Tags

- The HTML5 audio element works much like the video element
- Include the <audio> tag and a path that points to the audio file.
- You can modify the audio element using its control-related attributes, including:
  - autoplay
  - controls
  - loop

```
<audio src=”myaudio.mp3" controls="controls”></audio>
```

Audio Formats

- There are three primary formats of audio supported by Web browsers, including:
  - OGG
  - MP3
  - WAV
- Not every browser supports every audio type
  - The MP3 file format is the most commonly supported type
- You can use the `source` attribute to include multiple types

```
<audio controls>
     <source src="myaudio.mp3" type="audio/mp3"/>
     <source src="myaudio.ogg" type="audio/ogg"/>
</audio>
```

Summary

1	HTML
2	Basic Markup and Page Structure
3	Text Elements
4	Displaying Graphics
5	Canvas & SVG
6	Media in HTML5